a cura di Pierluigi Panunzi

# I «device driver»

terza parte

*Nelle ultime due puntate abbiamo iniziato ad analizzare le caratteristiche dei «device driver», a partire dalla loro descrizione e funzione principale di «estensione del DOS», per vedere poi le caratteristiche più interne quali il «Request Header» e le possibilità del DOS di attivare una particolare routine del driver stesso*

Ricordiamo brevemente le caratteristiche di un driver: si tratta di un insieme di routine attivabili esclusivamente da parte del DOS allorché quest'ultimo ne abbia bisogno ed in particolare l'attivazione avviene non attraverso una chiamata ad una di tante routine, ma semplicemente per mezzo di un unico entry point, subito dopo aver settato in maniera opportuna il già citato Request Header.

Da questo punto di vista dunque un driver è completamente diverso da un qualsiasi programma che viceversa siamo noi ad avviare digitandone da tastiera il nome od invocandolo tramite un batch oppure ancora attivandolo dall'interno di un altro programma: ripetiamo che viceversa rimane lì inattivo, finché il DOS non ne ha bisogno.

Ma quando ne ha bisogno?! È presto detto: quando ad esempio desideriamo effettuare operazioni sull'unità «S:» il cui gestore appunto, il driver, è proprio stato definito da noi, oppure quando desideriamo effettuare un output ad una certa unità «PRI:» che gestisce una stampantina seriale di tipo particolarissimo.

Infatti sappiamo che i device driver sono di due tipi, a seconda delle entità (item) con cui hanno a che fare: sarà perciò di tipo «character» oppure di tipo «block», nel quale ultimo caso invece del singolo carattere gestirà interi buffer di memoria (cioè un insieme di locazioni consecutive di memoria).

Abbiamo detto dunque più volte che un device driver consta di un certo numero di routine, ognuna rispondente ad un ben preciso compito, che il DOS si aspetta che la routine compia: va da sé che la routine di INPUT non dovrà mai effettuare altro che l'input di dati, altrimenti si potranno avere sicuri malfunzionamenti, oltre alla considerazione che non si è capito lo spirito di un device driver...

La scorsa puntata abbiamo già dato un elenco di quelle che sono le varie routine che formano un driver: per comodità di lettura anche in questa puntata riportiamo la tabellina (in figura 1) già vista.

Analogamente riteniamo utile ripubblicare un'altra tabellina, relativa al «Request Header» (vedasi la figura 2), in quanto fra breve ne torneremo a parlare in grande stile.

Prima però di passare ad analizzare i singoli comandi, ritorniamo un istante alla questione della chiamata da parte del DOS di una delle routine che compongono il driver, ovverosia l'attivazione di un comando del driver stesso.

## L'attivazione delle routine del driver

Abbiamo già parlato di questo argomento la scorsa puntata: ora cercheremo di essere un po' più chiari e concreti.

Supponiamo perciò che il DOS voglia

*Figura 1*
*Abbiamo riportato in questa tabella tutte le possibili funzioni contenute all'interno di un device driver, a seconda se quest'ultimo sia di tipo «character» oppure di tipo «block», con in corrispondenza il valore assunto dal campo «COMMAND CODE» che appare nel Request Header.*

| codice | funzione | |
|---|---|---|
| | block device | character device |
| 00 | INIT | INIT |
| 01 | MEDIA CHECK | NOP |
| 02 | BUILD BPB | NOP |
| 03 | IOCTL input | IOCTL input |
| 04 | INPUT | INPUT |
| 05 | NOP | NONDESTRUCTIVE INPUT |
| 06 | NOP | INPUT STATUS |
| 07 | NOP | INPUT FLUSH |
| 08 | OUTPUT | OUTPUT |
| 09 | OUTPUT with VERIFY | OUTPUT with VERIFY |
| 0A | NOP | OUTPUT STATUS |
| 0B | NOP | OUTPUT FLUSH |
| 0C | IOCTL output | IOCTL output |
| 0D | DEVICE OPEN | DEVICE OPEN |
| 0E | DEVICE CLOSE | DEVICE CLOSE |
| 0F | REMOVABLE MEDIA | NOP |

attivare il driver e cioè voglia inizializzarlo appunto con la routine INIT (che vedremo in dettaglio tra breve): il «voglia» di cui sopra è ovviamente una nostra interpretazione antropomorfica del computer...

In realtà il DOS è forzato ad effettuare le chiamate al driver in momenti ben precisi: a maggior ragione nel caso della INIT, che deve comunque essere eseguita, anche se poi il driver per ipotesi non verrà mai più interpellato.

Dicevamo dunque che il DOS vuole eseguire il comando «INIT»: per comunicarlo al driver, deve riempire un'apposita tabella (detta «Request Header», come ben sappiamo), fornire al driver l'indirizzo effettivo di memoria di tale tabella per mezzo della coppia di registri ES:BX, ed infine dovrà effettuare una chiamata alla routine detta di «strategy», il cui indirizzo era posto (anche questo lo sappiamo) nell'«Header» del driver stesso (attenzione! nell'Header del driver, non nel Request Header!).

Il driver, non appena in fase di esecuzione, dovrà dunque salvare i valori contenuti in ES:BX in celle di memoria a lui ben note e poi rientrare al DOS per mezzo di una istruzione RET di tipo FAR (inter-segment).

Subito dopo sarà il DOS stesso ad attivare il driver effettuando una chiamata alla «interrupt routine»: il driver potrà ora eseguire il comando desiderato dal DOS, leggendo appunto il Request Header, di cui conosce bene l'indirizzo.

In particolare l'«interrupt routine» conterrà al suo interno un controllo se il tipo di comando richiesto dal DOS sia implementato oppure no e solo in caso positivo si avrà il salto alla routine corrispondente al comando richiesto.

Non tanto complicato, ma assai macchinoso, non c'è che dire...

*Figura 2. Struttura del Request Header.*

| indir. | tipo | campo |
|---|---|---|
| 00 | BYTE | LENGTH : lunghezza dell'RH compresa la parte variabile |
| 01 | BYTE | UNIT CODE : codice dell'unita' interpellata (solo per i "block device", mentre e' senza significato per i "character device") |
| 02 | BYTE | COMMAND CODE : codice di comando (vedasi la tabella 1) |
| 03 | WORD | STATUS: word di stato |
| 05 | 8 BYTE | area riservata per l'MSDOS |
| 0D | var. | zona contenente i dati relativi alla particolare funzione attivata |

## *Il comando principale: INIT*

Si tratta del comando (o meglio, della routine) attivato in ogni caso, e per primo, da parte del DOS: corrisponde ad un valore nullo per il «command code» e servirà ai progettisti del driver per settare in uno stato iniziale ben noto tutte le risorse del driver stesso (siano esse «soft», quali variabili, contatori, aree di memoria da azzerare, oppure «hard» e cioè circuiti integrati da resettare o da programmare in maniera opportuna).

Nel caso del comando INIT si avrà un Request Header in cui, a partire dalla cella di offset pari a 0DH, ci saranno alcune informazioni.

Facendo riferimento alla figura 3, vediamo che in particolare sono presenti quattro campi, oltre alla parte fissa che è quella di figura 2, che sono:

— il numero di unità («Number of Units») che compongono il driver, nel caso in cui sia di tipo «block», numero che deve essere fornito dal driver stesso: se ad esempio il nostro driver gestisce due unità di memoria virtuale, ecco che il valore 2 dovrà essere posto in tale campo. In tal modo il DOS saprà che tra le unità logiche di tipo «block» ce ne saranno due generate dal driver stesso: se il computer ha già due floppy disk (rispettivamente «A:» e «B:»), ecco che le nuove unità si chiameranno «C:» e «D:»;

— l'indirizzo finale della parte residente del codice che costituisce il driver vero e proprio.

Ricordiamo a tal proposito che si intende per «parte residente» di un certo programma, quella parte che non verrà mai in alcun modo toccata da parte del DOS e che perciò potrà rimanere «sana e salva» al suo posto: il tutto in contrapposizione ad un programma qualunque, che, alla fine dell'esecuzione con relativo ritorno al DOS, viene viceversa rico-

*Figura 3. Struttura del Request Header nel caso di chiamata alla routine INIT, di inizializzazione del driver.*

| Comando INIT (0) | | |
|---|---|---|
| indir. | tipo | campo |
| 00 | 13 BYTE | REQUEST HEADER |
| 0D | BYTE | Numero di unita' previste dal driver (solo per i "block device", mentre e' senza significato per i "character device") |
| 0E | DWORD | End address della parte residente del driver |
| 12 | DWORD | Puntatore alla linea di comando per lettura dei parametri<br>Puntatore al BPB |
| 16 | BYTE | Numero associato alla prima unita' gestita dal driver |

perto da un altro programma attivato successivamente.

Nel nostro caso dunque abbiamo da parte del DOS la garanzia di poter contare su un driver il cui codice non verrà sovrascritto da altri programmi: a tale scopo dunque il driver dovrà fornire in tale campo l'indirizzo finale della parte di programma da rendere residente, che intende gestire (solo nel caso «block»), e conseguentemente deve fornire l'indirizzo del BPB, mentre infine deve fornire in ogni caso l'indirizzo dell'ultima locazione di memoria da mantenere residente.

Nel caso in cui si abbia la presenza di uno o più «parametri» nella linea di comando di cui sopra, il driver dovrà nell'oramai consueta forma «offset: segment»;

— un puntatore con due significati: in input è l'indirizzo del buffer di memoria dove sono posti i parametri specificati all'atto dell'attivazione del driver, mentre viceversa in output deve essere fornito al DOS il puntatore al cosiddetto BPB (del quale parleremo in dettaglio nel seguito) anche questo in forma «offset: segment».

Per quanto riguarda i parametri, facciamo l'esempio dell'attivazione di un driver all'interno del file CONFIG.SYS, del tipo

device = mydriver 64k

dove appunto è indicato il nome «mydriver» ed è specificato un parametro «64k», che il driver stesso dovrà poi processare. È perciò in tale campo che appare il puntatore della zona di memoria dove è stata memorizzata la stringa «64k», che il driver potrà così leggere;

— il «drive number», infine, nel caso di «block device driver» è un valore fornito dal DOS (a partire dalla versione 3.10) ed indica il numero associato alla prima unità gestita dal driver, con la convenzione secondo la quale l'unità «A:» viene indicata con il valore 0, l'unità «B:» con il valore 1 e così via.

Per quanto riguarda la funzione INIT, ci sono da aggiungere altre considerazioni, tra le quali ricorderemo ciò che il driver deve compiere comunque all'atto dell'attivazione.

In particolare, come abbiamo visto, deve indicare al DOS quante unità logiovviamente provvedere a salvare il contenuto del terzo campo (per avere l'indirizzo della zona di memoria in cui troverà i parametri), prima di andare a sovrascriverci l'indirizzo del BPB.

Inoltre la routine INIT dovrà effettuare l'inizializzazione delle risorse soft e hard ed infine, prima di cedere il controllo al DOS con una RET di tipo FAR, dovrà settare oppurtunamente la «status word» che abbiamo incontrato la scorsa puntata, segnalando la presenza di eventuali errori.

Infatti nel malaugurato caso in cui si abbia la presenza di errori (ad esempio se accade che la memoria richiesta per il corretto funzionamento del driver non sia effettivamente disponibile, oppure per un qualsiasi altro «intoppo» che impedirebbe il corretto funzionamento del driver), si potrà addirittura porre un valore nullo come numero di unità gestite dal driver e porre come ultimo indirizzo della parte residente pari a «CS:0000».

Con tale valore infatti si comunica al DOS che l'indirizzo dell'ultima locazione da mantenere residente è quella il cui offset è 0 e cioè in pratica si rinuncia al driver stesso (ricordano i lettori che un driver NON inizia ad offset 100H, come un qualunque programma di tipo «.com», ma bensì inizia all'indirizzo 0, al quale deve essere posto l'«Header» ?!), evitando così il benché minimo spreco di memoria da parte del driver, che non potrebbe mai funzionare e viceversa rimarrebbe inutilizzabile in memoria.

| Indir | tipo | campo |
|---|---|---|
| 0 | WORD | numero di byte per settore |
| 2 | BYTE | numero di settori per allocation block |
| 3 | WORD | numero di settori riservati |
| 5 | BYTE | numero di FAT |
| 6 | WORD | numero massimo di elementi nella directory |
| 8 | WORD | numero totale di settori |
| 0A | BYTE | MEDIA DESCRIPTOR |
| 0B | WORD | numero di settori occupati da una FAT |

*Figura 4*
*Struttura del BPB («Bios Parameter Block»), nel quale sono indicate le caratteristiche peculiari dell'unità logica, nel caso in cui il driver sia di tipo «block».*

## *Il BPB («Bios Parameter Block»)*

Accenniamo in questa sede, visto che ne abbiamo parlato parecchio, al significato ed alla struttura del BPB: si tratta di una tabella (che vediamo in figura 4), contenente parecchie informazioni utili per poter gestire correttamente un dispositivo logico di tipo «block».

In particolare vediamo che nel BPB appaiono parecchi campi, il cui contenuto è il seguente:

— il primo campo indica il numero di byte per settore;

— il secondo indica invece il numero di settori che compongono un blocco di allocazione (che rappresenta l'insieme fondamentale su cui lavora il DOS): tale valore deve necessariamente essere una potenza di 2;

— quindi abbiamo il numero di settori riservati (dove in genere viene posto il sistema, ma che viceversa possono essere quelli appartenenti ad un'altra partizione, ma qui il discorso si complica oltre le intenzioni e perciò l'abbandoneremo subito, come non detto...);

— successivamente abbiamo il numero di FAT presenti nell'unità: ricordiamo che per FAT («File Allocation Table») si intende una tabella presente in tutti i dischi (floppy, minidisk ed hard-disk) che indica, in maniera alquanto complessa, in quali punti del disco stesso sono stati allocati tutti i file presenti.

Per motivi di sicurezza e cioè per conservare l'integrità delle informazioni presenti sul disco, di solito la FAT è riportata in duplice copia, anche se ciò comporta l'utilizzazione di una parte iniziale del dischetto.

— Poi è presente il numero massimo di «directory entry» e cioè di «voci» all'interno della directory, intendendo con tale termine, oltre ai file veri e propri, anche gli elementi che identificano una sub-directory.

— Successivamente compare un valore che indica il numero totale di settori che compariranno nell'unità logica, includendo nel conteggio anche il settore di «bootstrap», i settori dedicati alla directory e quelli riservati per la FAT.

— Il penultimo campo rappresenta il «Media Descriptor», in pratica un valore che identifica le caratteristiche dell'unità logica in esame: di tale byte parleremo ampiamente nella prossima puntata.

— Infine è presente il numero di settori occupati da una singola FAT.

Con la descrizione del BPB chiudiamo dunque la puntata: nella prossima analizzeremo altri comandi relativi ad altrettante routine da implementare in un device driver.

# POSTAL COMPUTER

## OFFERTA SPECIALE: RAM

**PC XT IBM COMPATIBILE**
**L. 750.000**
SCHEDA MADRE 6/10 MHZ, 1 DRIVE 360K, SCHEDA CGA O HERCULUS, 256K ESPANDIBILE A 640K SU PIASTRA, TASTIERA AVANZATA 101 TASTI

**PC XT IBM COMPATIBILE**
**L. 1.200.000**
SCHEDA MADRE 6/10 MHZ, 1 DRIVE 360K, SCHEDA GRAFICA HERCULUS O CGA, 1 HARD DISK 20 MEGA, 256 ESPANDIBILE A 640K SU PIASTRA, TASTIERA AVANZATA 101 TASTI.

**PC AT IBM COMPATIBILE**
**L. 2.600.000**
SCHEDA MADRE 80286 12 MHZ, 0 WAIT, 512 K ESPANDIBILE A 1024 K, 1 DRIVE 5,25'' DA 1,2 MBYTE 1 WINCHESTER DA 40 MBYTE 20MS, SCHEDA SUPER EGA, TASTIERA AVANZATA 101 TASTI.

**386 TOWER 16/20 MHZ**
**L. 5.750.000**
MICROPROCESSORE 80286 16/20 MHZ 0 WAIT RAM 2 MB (80 NS) ESPANDIBILE A 16 MB, 8 SLOT, SCHEDA EGA, 1 DRIVE DA 1,2 MB 1 DRIVE 3.5 720 KB, WINCHESTER DA 40 MB.

**PREZZI SU RICHIESTA**

| | |
|---|---|
| **COVERTASTIERA** 84 TASTI | 15.000 |
| **COVERTASTIERA** 101 TASTI | 20.000 |

**TUTTI I NS. PREZZI SONO IVA 19% ESCLUSA, SPESE DI SPEDIZIONE ESCLUSE**

*** CASSETTE VHS MASTER ***
- HG E 120 L. 4.600 - HG E 180 L. 5.450

## TELEFAX MURATA M-1
## L. 1.500.000

- COMPATIBILITÀ: G2 G3
- VELOCITÀ DI TRASMISSIONE 15 SECONDI
- APPARECCHIO TELEFONICO A TASTIERA INCORPORATO
- FOTOCOPIATORE
- RICEZIONE AUTOMATICA
- ROTOLO CARTA TERMICA 216 mm x 30 metri.
- OROLOGIO/CALENDARIO DIGITALE

**GARANZIA 18 MESI**

| | |
|---|---|
| HARD DISK SEAGATE 20 MB | L. 350.000 |
| HARD DISK CONTROLDATA 40 MB | L. 680.000 |
| HARD DISK CONTROLLER PER XT | L. 100.000 |
| HARD DISK CONTROLLER PER AT | L. 220.000 |
| SCHEDA GRAFICA SUPER E.G.A. | L. 300.000 |
| SCHEDA MULTI I/0 | L. 110.000 |
| SCHEDA SERIALE | L. 40.000 |
| SCHEDA PARALLELA | L. 35.000 |
| SCHEDA PORTA JOYSTICK | L. 28.000 |
| SCHEDA MADRE XT | L. 190.000 |
| SCHEDA MADRE AT (12 MHZ 0 WAIT) | L. 650.000 |
| TASTIERA AVANZATA 101 TASTI | L. 110.000 |
| DRIVE 5,25 360KB | L. 140.000 |
| DRIVE 5,25 1,2MB | L. 190.000 |
| DRIVE 3,50 720KB | L. 190.000 |
| DRIVE CONTROLLER | L. 49.000 |
| CAVO PARALLELO | L. 15.000 |
| DATA SWITCH A 2 PORTE | L. 60.000 |
| MOUSE ANKO | L. 59.000 |
| JOYSTICK I.B.M. ANKO | L. 45.000 |

## STAMPANTI CITIZEN GRAFICA - NLQ

| | |
|---|---|
| **CITIZEN 120 D L. 360.000**<br>120 CPS, SET. EPSON IBM 80 COL. TRATO IN TRAZIONE, FRIZIONE INTER. OPZIONALE IBM/COMMODORE | **CITIZEN MSP 50**<br>L. 1050.000<br>250/300 CAR/SEC., 80 COL. |
| **CITIZEN LSP 100**<br>L. 550.000<br>-160 cps, 80 COL. | **CITIZEN MSP 55**<br>L. 1.230.000<br>250/300 CAR/SEC, 136 COL. |
| **CITIZEN MSP 10E**<br>L. 650.000<br>- 160 CAR/SEC, 80 COL. | **CITIZEN HQP 40**<br>L. 1.160.000<br>- 24 AGHI, 200 CPS ALTISSIMA QUALITÀ |
| **CITIZEN MSP 15E**<br>L. 680.000<br>160 CAR/SEC, 136 COL. | **CITIZEN HQP 45**<br>L. 1.530.000<br>- 24 AGHI, 200 CPS ALTISSIMA QUALITÀ |
| **CITIZEN MSP 40**<br>L. 775.000<br>- 200/240 CAR/SEC, 136 COL. | **CITIZEN PREMIERE 35**<br>L. 1.250.000<br>- MARGHERITA PROFESSIONALE, 35 CPS |
| **CITIZEN MSP 45**<br>L. 950.000<br>- 200/240 CAR/SEC, 136 COL. | **CITIZEN OVERTURE 110**<br>* L. 3.600.000<br>- STAMPANTE LASER |

**TUTTI I PRODOTTI CITIZIEN SONO COPERTI DA CERTIFICATO DI GARANZIA DELLA VALIDITÀ DI DUE ANNI**

| | |
|---|---|
| **PORTA FLOPPY** 50/60 3 1/2 | L. 15.000 |
| **PORTA FLOPPY** 100 3 1/2 | L. 20.000 |
| **PORTA FLOPPY** 40/50 5 1/4 | L. 18.000 |
| **PORTA FLOPPY** 80/90 5 1/4 | L. 23.000 |

| | |
|---|---|
| **MONITOR 12'' TTL** L. 150.000 | F/V |
| **MONITOR 12'' COMPOSITO** L. 150.000 | Ambra |
| **MONITOR DUAL 12''** L. 200.000 | F/V |
| **MONITOR A COLORE MULTITECH** L. 555.000 | colore ambra F/V |
| **MONITOR PHILIPS COL. 8833** L. 500.000 | colore |

* PRODOTTI COMMODORE SU RICHIESTA *

## DISCHETTI OFFERTA SPECIALE

| BULK | 10 | 100 | 500 |
|---|---|---|---|
| 5 1/4 DS DD | 950 | 850 | 750 |
| 5 1/4 HD | 2200 | 2100 | 2000 |
| 3 1/2 DS DD | 1900 | 1800 | 1700 |

| NASHUA | 10 | 100 | 500 |
|---|---|---|---|
| 5 1/4 DS DD | 1400 | 1300 | 1200 |
| 5 1/4 HD | 2500 | 2400 | 2300 |
| 3 1/2 DS DD | 2200 | 2000 | 1900 |

**SU TUTTI I NOSTRI PRODOTTI MAGNETICI OFFRIAMO IL NOSTRO SERVIZIO DI SOSTITUZIONE IMMEDIATA DEI PEZZI DIFETTOSI**

**PREZZI IVA 19% ESCLUSA**

**TEL. 06/3652427/3652431** **TELEFONATECI**